Veli-Matti Assila

Langaton verkkoyhteys ja palvelin tutkimusautolle

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Luonnontieteiden ala
Tietojenkäsittelyn koulutusohjelma
Kevät 2012

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ
TIIVISTELMÄ

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Veli-Matti Assila	
Työn nimi Langaton verkkoyhteys ja palvelin tutkimusautolle	
Vaihtoehtoiset ammattiopinnot Järjestelmän ylläpito	Ohjaaja(t) Joona Tolonen
	Toimeksiantaja Kajaanin ammattikorkeakoulu
Aika 18.5.2012	Sivumäärä ja liitteet 35

Tämän opinnäytetyön aiheena on rakentaa Kajaanin ammattikorkeakoululla sijaitsevalle tutkimusautolle uusi tietokantapalvelin vanhan kehityspalvelimen tilalle. Vanha palvelin korvataan uudella virtualisointiratkaisulla, joka mahdollistaa palvelimen jatkokehityksen myös tulevaisuudessa.

Uuteen palvelimeen asennetaan vanhaa palvelinta vastaavat ohjelmistot, jonka jälkeen palvelin sisältää WWW- ja SQL-palvelimet. Työhön liittyy myös tietoturva, jonka tärkeys kohoaa palvelimen ollessa tuotantokäytössä oleva palvelin.

Teoriaosuus käsittelee pilviteknologiaa, sen määritelmää ja mahdollisuuksia myös perinteisen IT:n ulkopuolella. Käytännön osuus keskittyy palvelimen asennuksen etenemisen kuvaukseen ja ongelmatilanteiden ratkaisujen muodostumiseen.

Kieli	Suomi
Asiasanat	Linux, palvelimet, tietokannat, tietoturva
Säilytyspaikka	☒ Verkkokirjasto Theseus ☒ Kajaanin ammattikorkeakoulun kirjasto

KAJAANIN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES

THESIS
ABSTRACT

School Business	Degree Programme Business Information Technology
Author(s) Veli-Matti Assila	
Title Wireless Connection and a Server for a Research Car	
Optional Professional Studies System Administration	Instructor(s) Joona Tolonen
	Commissioned by Kajaani University of Applied Sciences
Date 18.5.2012	Total Number of Pages and Appendices 35

The purpose of this thesis was to build a new database server which replaces an old development server for a research car located in Kajaani University of Applied Sciences. The old server is replaced by a new virtualization solution which allows further development of the server in future.

The same software packages are installed to the new server as are in the old one, which after the new server includes WWW- and SQL-servers. Data security is also a part of the project when the server serves in the production environment.

The theoretical part of the thesis covers cloud computing technology: its definition and possibilities for usage also outside of traditional IT. The practical part focuses on describing the progress of installing the server software and forming solutions in problem situations.

Language of Thesis	Finnish
Keywords	Linux, servers, databases, information security
Deposited at	☒ Electronic library Theseus ☒ Library of Kajaani University of Applied Sciences

SISÄLLYS

SYMBOLILUETTELO

Lähiverkko/LAN	Local area network. Yksittäisten työasemien ja palvelinten muodostama yksityinen verkko, jossa siirretään tietoa ja jaetaan palveluita. Internet ei ole lähiverkko.
IP-osoite	Osoite, jolla yksilöidään verkossa olevat laitteet. IP-osoitteen avulla datapaketit löytävät perille kohteeseensa.
WLAN	Wireless local area network, langaton lähiverkko.
Palomuuri	Ohjelmisto tai laite, joka suojaa laitteita verkosta tulevilta hyökkäyksiltä ja hallitsee verkkoyhteyttä haittojen varalta.
Pilvipalvelu	Verkossa sijaitseva palvelu, jossa palvelin toimii tiedon säilyttäjänä ja mahdollisesti myös sovellusten suorittajana.
DMZ	Demilitarized zone. Suojaamaton verkkoalue, jossa olevia laitteita ei suojata tietoverkon suojausjärjestelmillä, kuten palomuurilla.
Avoin ja suljettu lähdekoodi	Ohjelmiston rakenteena oleva koodi, joka avoimena on kaikkien vapaasti nähtävillä ja usein myös halutessaan muokattavana, kun taas suljettuna koodi pysyy vain tekijänsä tietona eikä kolmas osapuoli saa sitä muokata.
Linux	Avoimen lähdekoodin käyttöjärjestelmäydin, joka perustuu Unix-ytimeen.
Windows	Microsoftin valmistama kaupallinen käyttöjärjestelmä.
Mac OS X	Applen valmistama Unix-pohjainen käyttöjärjestelmä, jota myydään ainoastaan Applen itse valmistamilla laitteilla käytettäväksi.
Debian	Tuhansien vapaaehtoisten rakentama ilmainen Linux-pohjainen käyttöjärjestelmä.

Graafinen käyttöliittymä	GUI, eli graphical user interface. Käyttöliittymä, jossa järjestelmän toiminnot esitetään graafisesti kuvilla ja sitä voidaan käyttää esimerkiksi hiirellä.
Komentorivikäyttöliittymä	Käyttöliittymä, jossa järjestelmää käytetään komentorivillä jolloin kaikki toiminnot suoritetaan näppäimistöllä kirjoitettavilla komennoilla.
Kernel/ydin	Käyttöjärjestelmän alin osa, joka toimii runkona kaikelle tietokoneessa tapahtuvalle ohjelmistotoiminnalle.
LAMP	Linux Apache MySQL PHP. Ohjelmistopaketti.
Apache	HTTP palvelinohjelmisto.
PHP	Dynaaminen ohjelmointikieli, jota voi käyttää toiminnallisuuden luomiseen www-sivustoille.
MySQL	Tietokantaohjelmisto. Ylläpitää ja luo tietokantoja.
Pear	PHP:n ympärille rakennettu runko- ja julkaisujärjestelmä.
GD	Kuvanmuokkausohjelma. Toimii palvelimella kuvien käsittelijänä.
DNS	Domain Name System, eli nimipalvelujärjestelmä. Sen avulla selvitetään tekstimuodossa olevien Internet-osoitteiden IP-osoite.
FTP	File Transfer Protocol. Tiedonsiirtoprotokolla.
NFS	Network File System. Protokolla, jonka avulla järjestelmä voi käyttää verkossa olevaa tallennustilaa.
TCP	Tietoliikenneprotokolla, jossa tarkistetaan bittien perille pääsy ja niiden oikeassa järjestyksessä oleminen.

UDP	Tietoliikenneprotokolla jossa ei vaadita pysyvää yhteyttä lähettäjän ja vastaanottajan välille. Tällöin bittien vastaanottoa ei varmisteta.

1 JOHDANTO

Tämän opinnäytetyön aiheena on luoda verkkoyhteys ja palvelinjärjestelmä Kajaanin ammattikorkeakoulun omistamaa, tutkimuskäyttöön tarkoitettua ajoneuvoa varten. Järjestelmää tullaan käyttämään autosta kerätyn tiedon tallentamiseen ja tavoitteena on myös järjestelmän kaupallinen hyödyntäminen. Järjestelmä tulee osaksi Kajaanin ammattikorkeakoulun omaa IT-järjestelmää.

Alkutilanteessa ajoneuvo siirtää tietoa erillisen kannettavan tietokoneen avulla. Tämä tietokone siirtää tietoja koulun alueella olevan pienen, erillisen WLAN-verkon kautta ja vastaanottavana palvelimena toimii yksittäinen työasema. Tämä järjestelmä halutaan siis uusia.

Nykyisessä järjestelmässä on useita puutteita. Ensinnäkin ajoneuvossa käytettävä yhteystapa on kömpelö ja se halutaan automatisoida käyttäen hyödyksi WLAN:n lisäksi myös matkapuhelindataverkkoja. Vastaanottava palvelin on nykyisellään riittämätön, sillä se toimii pääsääntöisesti vain tiedon säilyttäjänä, mutta vastaavasti jatkokäyttö ja tulevat järjestelmän laajennukset saattaisivat olla sille liian suuria. Ajoneuvon käyttötietoa keräävä tietojärjestelmä itsessäänkin on vielä käytännössä keskeneräinen ja sen jatkokehitykselle halutaan antaa mahdollisuudet myös palvelinpuolella.

2 PILVIPALVELU

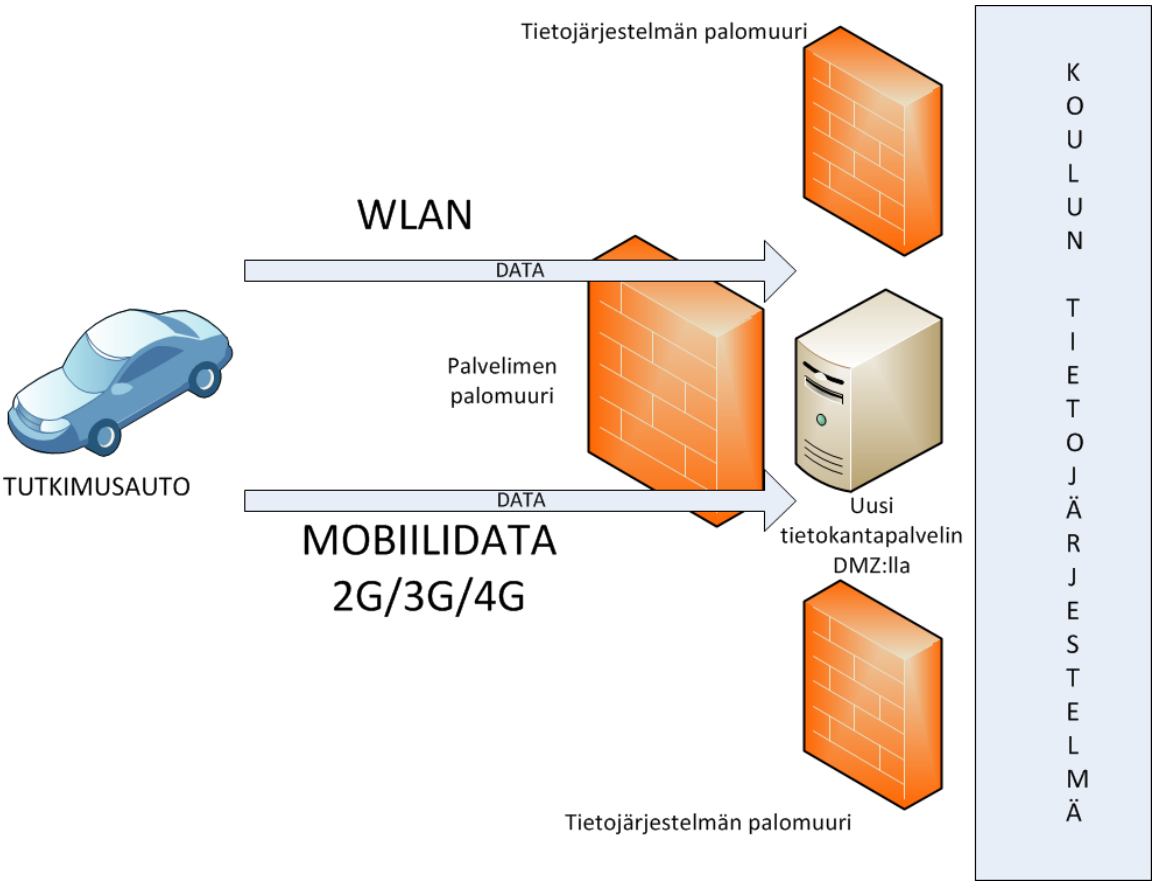
Pilvipalvelut ovat verkkoyhteyksien kehittyessä mahdollistunut teknologia, jossa tietoa ei enää tarvitse suorittaa ja säilyttää paikallisesti (Mell & Grance 2011, 2). Teknologia sallii laajenevan laitekirjon välisen jatkuvan vuorovaikutuksen ja sitä tuodaan jatkuvasti uusiin alustoihin. Autot ovat yksi pilviteknologian mahdollistama uusi alusta.

Pilviteknologia ja verkkoyhteydet mahdollistavat ajoneuvoihin karkeasti jakaen kaksi lisätoiminnallisuutta. Matkustajille ne näkyvät oman digitaalisen elämän integroitumisella ajoneuvoihin verkkopalvelujen ja median käyttönä, jolloin esimerkiksi musiikkia on mahdollista kuunnella verkon kautta omalta kotona sijaitsevalta tietokoneelta, sekä ympäröivän maailman tiedon saatavuutena, kuten esimerkiksi ruuhkatietojen päivittymisenä navigaattoriin. Toinen merkittävä toiminnallisuus on ajoneuvon sisäisen tiedon jatkokäyttö, jolloin esimerkiksi statistiikkaa voidaan lähettää reaaliaikaisesti palvelimelle. Tätä tietoa hyväksikäyttäen kuljettajalle voidaan esimerkiksi ilmoittaa jo ajon aikana mahdollisista ongelmista ja vaaratilanteista tai auttaa ajotavan muuttamista kulutuksen vähentämiseksi. Pilviteknologialla pyritään siis parantamaan matkan viihtyvyyttä ja sujuvuutta, mutta myös hyödyntämään ajonaikaista tietoa tehokkaammin. (Squatriglia 2012.)

Pilviteknologia yleistyy ajoneuvoissa nopeasti ja lähes kaikissa uusissa ajoneuvoissa on pilviteknologiaksi määriteltävä toiminnallisuus, kuten esimerkiksi navigaattori, joka noutaa reaaliaikaista tietoa valmistajansa palvelimelta. Teknologia on liikenneteollisuudessa kuitenkin vielä niin uusi ettei täysipainoista hyödyntämistä vielä ole, mutta uusia tehokkaammin pilviteknologiaa käyttäviä konseptiautoja valmistetaan jatkuvasti. Esimerkkinä tästä on Fordin valmistama Evos, jossa oman älypuhelimen voi yhdistää ajoneuvoon ja tällöin sekä auto että puhelin kommunikoivat keskenään ja muiden verkossa olevien ajoneuvojen kanssa. (Emspak 2011.)

Ehkä pisimmälle tietoa käsittelevien ajoneuvojen käytössä on päässyt Google, joka on vuosia rakentanut itsestään ajavaa autoa. Tässä korostuu tiedon käsittelyn tarkkuus ja kommunikointi ympäristön kanssa, sillä vahinkotilanteessa tuhot voivat olla suuret. Ajoneuvoa on testattu satoja tuhansia kilometrejä ja tulokset ovat olleet lupaavia. (Markoff 2010.)

Opinnäytetyössä tehtävä palvelin on toiminnaltaan yksinkertainen pilvipalvelin, joka tarjoaa ajoneuvon tiedolle säilytys- ja jatkoprosessointitilan. Yksinkertaisen palvelimesta tekee se, ettei tietoa siirry vuorovaikutteisesti vaan ainoastaan ajoneuvo kerää dataa ja palvelin ainoastaan vastaanottaa sen. Se, että toisena osapuolena on ajoneuvo, ei itse asiassa aiheuta mitään erityisiä vaatimuksia palvelimelle, koska tiedonsiirrossa käytetään TCP/IP-protokollaa. Tiedonsiirto ei siten eroa esimerkiksi tietokoneen ja palvelimen välisestä tiedonsiirrosta mitenkään. Ainoa huomioitava ero on yhteystyypissä, sillä langaton matkapuhelinverkko sisältää ominaisuuksia, jotka täytyy huomioida palvelinta muokatessa. Kuviossa 1 on nähtävissä valmiin toteutuksen suunniteltu rakenne.



Kuvio 1. Uutta järjestelmää kuvaava kaaviokuva.

2.1 Mobiilidata

Vaikka ajoneuvo itsessään ei luo erityisiä vaatimuksia yhteydelle, langaton tiedonsiirto saattaa niin tehdä. Paikallinen WLAN-verkko toimii ilman erityisvaatimuksia vaaditulla laitteistolla, mutta samalla sen toiminta-alue on pieni, eikä siten riittävä liikkuvaan käyttöön. Liikkuvaa käyttötarkoitusta varten tutkimusautossa on käytössä myös mobiilidata, joka toimii tarvittaessa globaalisti, mutta toisaalta tuo vaatimuksia yhteydelle ja palvelimen asetuksiin.

Mobiilidata, josta käytetään yleisesti termejä 2G, 3G ja 4G, toimii matkapuhelinoperaattorin kautta käyttäen tehtävään suunniteltua modeemia ja tavallista SIM-korttia. Yhteys muodostetaan operaattorin tukiasemien kautta ja siten yhteys voidaan muodostaa ja ylläpitää siellä missä on saatavana operaattorin verkko. Yhteyden nopeus ja tyyppi riippuvat tukiasemassa olevasta tekniikasta ja käytössä olevasta modeemista. (Unuth 2012.)

Kattavuuden ansiosta mobiilidata soveltuu hyvin liikkuvaan tiedonsiirtoon ja siten myös ajoneuvoille. Katvealueet, ja siten myös yhteyskatkot, ovat mobiilidatallakin mahdollisia, mutta yhteys voidaan verkon löytyessä muodostaa uudelleen ja siten yhteyskatko on mahdollista pitää mahdollisimman lyhyenä. Maantieteellisesti verkot ovat Suomessa kuitenkin laajoja ja yhteys on teoriassa mahdollista pitää yllä lähes joka paikassa. (Viestintävirasto 2011.)

Mobiilidata vaatii tiettyjen asioiden tiedostamista. Ensinnäkin mobiilidataan kuuluva IP-osoite ei yleensä ole kiinteä, vaan joka kerta kun yhteys menee poikki ja muodostuu uudestaan myös IP-osoite vaihtuu. Käytännössä tämä tarkoittaa sitä, ettei esimerkiksi palomuurimäärityksissä voida tehdä IP-sidonnaisia sääntöjä, sillä kummankin osapuolen IP-osoitteita ei voida varmasti tietää. Se, onko mobiilidatalla mahdollista saada kiinteä IP-osoite, riippuu operaattorista.

Toiseksi pitää muistaa, ettei mobiilidata ole aina vakaa tai nopea. Yhteyden nopeuteen vaikuttaa suuresti verkossa oleva käyttäjämäärä ja käyttäjien määrän ylittäessä tukiasemien kapasiteetin voivat yhteyden nopeus ja vakaus heikentyä huomattavasti (Repo 2011).

Mahdollisiin katkoihin voi varautua tutkimusautossa esimerkiksi puskuroimalla dataa katkojen ajan ja yhteyden palattua datan siirtoa voi jatkaa.

Mobiilidata on kaikista puutteistaan huolimatta ensimmäinen varteenotettava teknologia Internet-yhteyksien tuomiseksi liikkuviin kohteisiin. Yhteyden nopeus ja verkkoviive voivat parhaimmillaan vastata jopa kiinteitä, johtoja pitkin toimivia Internet-yhteyksiä ja täten mobiilidatalla pystyy toteuttamaan aiemmin kiinteällä yhteydellä toteutettuja palveluja. (White Paper 2009.)

3 PALVELIMEN RAKENTAMINEN

Ajoneuvoa varten luodaan palvelinjärjestelmä. Se tulee osaksi Kajaanin ammattikorkeakoulun tietojärjestelmää DMZ-alueelle ja toimii Internetin välityksellä myös koulun oman lähiverkon ulkopuolella. Ajoneuvo tulee olemaan yhteydessä palvelimeen langattomia verkkoteknologioita käyttäen. Alkuperäinen palvelin rakennettiin testikäyttöiseksi kehitysympäristöksi, joka yksittäisenä työasemana jäisi tuotantokäytössä epäkäytännölliseksi ja tästä syystä varsinainen tuotantopalvelin rakennetaan koulun muun tietojärjestelmän yhteyteen.

Palvelin asennetaan käyttäen virtualisointiteknologiaa, joka tarkoittaa sitä, että palvelinta varten ei ole omaa erillistä laitteistoa vaan se toimii ohjelmallisesti luodulla laitteistolla. Käytännössä tämä tarkoittaa että on olemassa keskitetty, tehokas laitteisto, jolle on asennettu tarvittava virtualisointiympäristö, joka taas vastaavasti ohjaa ja luo sen alaisuudessa toimivia ohjelmallisia tietokoneita. Tällä tavoin voidaan yhdelle laitteistolle luoda useita eri tehoisia laitekokonaisuuksia, jotka keskenään jakavat fyysisten laitteiden resurssit. Virtualisointia varten koululla on käytössä VMwaren valmistama vSphere, jolla virtualisointiympäristöjä voidaan rakentaa. Sitä hallitaan käyttöjärjestelmälaboratorio-luokan koneilla olevasta vSphere Client-ohjelmasta.

Palvelinalustaksi on mahdollista valita teoriassa mikä tahansa markkinoilla olevista ratkaisuista. Vaihtoehdot rajoittuvat käytännön syistä kuitenkin kolmeen, ylivoimaisesti suurimpaan vaihtoehtoon: Linuxiin, Windowsiin ja Mac OS X:ään (Pettey 2011). Koulun järjestelmässä on käytössä kaikkia kolmea edellä mainittua, eli Windows-, Mac OS X- ja Linux-palvelimia. Näistä kaksi ensimmäistä ovat kaupallisia tuotteita, eli näiden hankkimiseen ja tarvittaessa myös käyttöön tarvittaviin lisensseihin tarvitsee rahaa. Tämä syy osaltaan vaikuttaa siihen että lopulliseen palveluun valittiin alustaksi Linux. Toinen merkittävä tekijä on Linuxin maine tietoturvallisena alustana (Hess 2010).

Windowsia ei alustaksi valittu useasta syystä. Vaikka Windows on markkinajohtaja, on se myös maksullinen, haittaohjelmille altis ja tarpeeseen liian raskas. Rakennettava palvelu ei sinällään vaadi tehokasta laitteistoa, jolloin raskas käyttöjärjestelmä on turha rasite. Windows-palvelimen voi asentaa kevyempänä komentoriviversiona, mutta tämä ei oikeasti

ole kevyt, sillä Windows asentaa silti graafisen käyttöliittymän (Petri 2009). Graafinen käyttöjärjestelmä on laitteistoa paljon kuormittava, joten järjestelmä halutaan asentaa komentorivipohjaisena.

Mac OS X-palvelimen valintaa puolustaisi se, että alkutilanteessa tutkimusauton palvelimena toimii Mac OS X-alustalla toimiva tietokone. Uuden järjestelmän rakentaminen samalle käyttöjärjestelmällä vanhan palvelimen kanssa takaisi, että palvelu olisi helppo siirtää uuteen alustaan. Palvelinohjelmisto toimii kuitenkin vain Applen itse valmistamilla laitteistoilla, joita koulussa on vain vähän, joten valinta rajoittaisi paljon mahdollista palvelun siirtoa uudelle laitteistolle tulevaisuudessa. Myös se, että koulun Mac OS X-palvelin palvelee jo muuta käyttötarkoitusta estää käytännössä palvelun rakentamisen kyseiselle alustalle.

3.1 Linux

Linux sanana tarkoittaa suomalaisen Linus Torvaldsin kehittämää Unix-pohjaista käyttöjärjestelmäydintä. Hän teki ytimestä vapaasti levitettävän, eli jokainen voi halutessaan käyttää sitä omiin tarkoituksiinsa. Avoimuuden ansiosta Linuxista on kehittynyt laaja käyttöjärjestelmäperhe ja sitä on saatavilla lukemattomissa eri distroissa. (Linux.org 2012.)

Linuxia jaetaan siis jakeluversioina, eli distroina. Suuri osa näistä on ilmaisia ja vapaalla lähdekoodilla, mutta myös kaupallisia ratkaisuja on tarjolla. Tunnettuja distroja ovat muun muassa:

- Debian
- Ubuntu
- Red Hat
- SuSE ja OpenSuSE

Myös monet matkapuhelinkäyttöjärjestelmät ovat jalostuneet Linuxista, tunnetuimpana luultavasti Android. (Torikka 2009, 14.)

3.2 Debian GNU/Linux

Linux on saatavilla useina eri jakeluina ja osan niistä ollessa suunniteltu useaan eri tarkoitukseen voi osa olla tarkoitettu johonkin erityiseen tehtävään. Siksi on hyvä tutkia etukäteen mikä distro on paras tarvittavaan käyttötarkoitukseen. (Tuxradar.com 2009.)

Tutkimusautoa varten ei tarvita mitään erityistä jakelua, joten valinta päätettiin kohdistaa tukisyistä yleisimpiin ja tunnetuimpiin jakeluihin. Näitä ovat esimerkiksi Ubuntu, Debian, Red Hat ja SuSE. Valinta kohdistui Debianiin, sillä se on tunnettu vakaudestaan, tietoturvastaan ja on tarvittaessa kohtuullisen kevyt, eli ei vaadi tehokasta laitteistoa (Byfield 2009). Ubuntu on jalostettu Debianista ja on suunniteltu tavalliselle kotikäyttäjälle. Se sisältää osia, jotka olisivat tarpeettomia palvelinta ajatellen. Red Hat ja SuSE taas ovat monilta osin suunnattu yrityksille, joten tutkimusauton käyttötarpeille ne ovat tarpeettoman raskaita. Debianin valintaa puolustaa myös laaja yhteisöllinen tuki ja ohjeistus, jota löytää helposti esimerkiksi Internetin hakukoneilla.

Debiania on saatavilla usealle eri alustalle, mutta yleisesti käytetään i386- ja amd64-prosessorien työpöytä- ja palvelinversioita. Versiot eroavat toisistaan hieman ytimen osalta palvelinversion sisältäessä tuen laajemmalle valikoimalle laitteistojen ominaisuuksia sekä mukana tulevien ohjelmistojen osalta. (DifferenceBetween.net 2012.)

4 PALVELIN PYSTYYN

Ennen palvelimen asennusta tulee pohtia paria asiaa: Mitä sovelluksia tarvitaan, kuinka paljon tehoa laitteistossa tarvitaan ja mitä vaatimuksia palvelin aiheuttaa muuhun infrastruktuuriin? (Venezia 2010.)

Tarvittavat sovellukset sovitaan opinnäytetyön tilaajan kanssa, tässä tapauksessa koulun kanssa. Tutkimusautolla on jo käytössään erillinen palvelin, joka tässä työssä on tarkoitus korvata, ja uuden palvelimen sisältämät palvelut tulevat olemaan samat kuin vanhassa palvelimessa olevat. Samat sovellukset sallivat vanhojen palvelujen kohtuullisen helpon siirtämisen uuteen alustaan. Tarvittavat sovellukset ovat LAMP, Pear sekä GD.

Palvelin asennetaan virtuaaliseksi, jolloin suorituskyvyn skaalaaminen jälkikäteen on mahdollista ja helppoa. Palvelin ei tule vaatimaan paljoa tehoa ja siksi virtuaalinen laitteisto jätetään nykymittapuulla vaatimattomaksi yhdellä prosessorilla ja 512 megatavun muistilla. Laitteiston levyn koko jätetään myös pieneksi, mutta sallitaan koon kasvattaminen tarpeen mukaan.

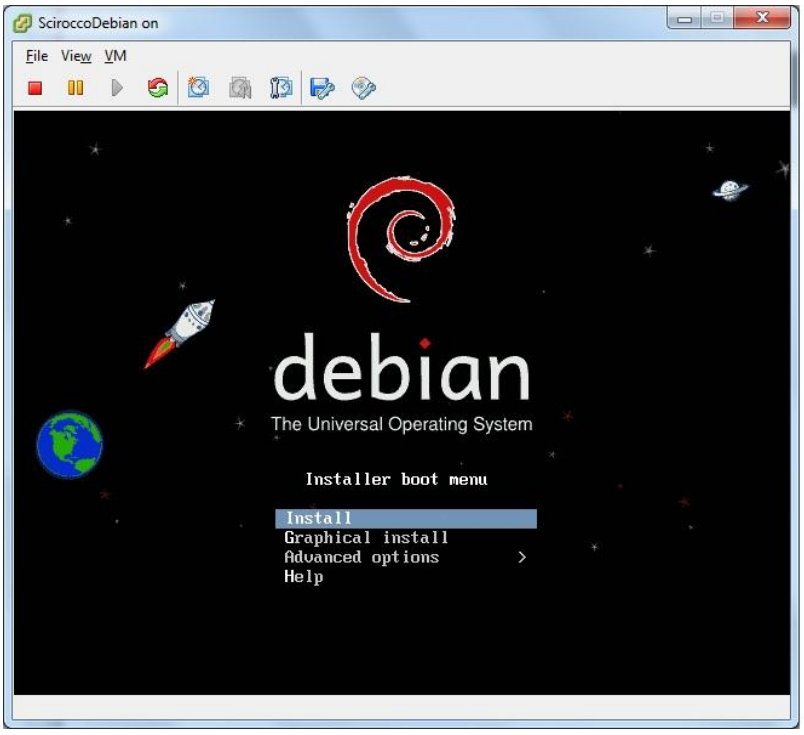
Palvelinta tullaan tarvitsemaan koulun oman sisäisen verkon ulkopuolelta, jolloin verkkoarkkitehtuuri vaatii vähän konfigurointia. Se tullaan sijoittamaan DMZ-alueelle, jolloin palvelimelle tehdään erikseen oma palomuuri. Tarvitaan myös oma julkinen IP-osoite, jonka avulla palvelimeen saa yhteyden myös yleisistä verkoista. Kaikki tämä pitää rakentaa käsi kädessä riittävän tietoturvan kanssa, sillä juuri verkko aiheuttaa suurimmat tietoturvariskit palvelimille (Smith 2009, 470).

4.1 Perusasennus

Virtuaalikoneen asennus alkaa aina laitteiston määrittämisellä. Koska tuleva palvelin ei tarvitse tehokasta laitteistoa, hyväksytään vSpheren tarjoamat oletusasetukset käytännössä sellaisinaan. Nämä asetukset ovat: yksi yhden gigahertzin prosessoriydin, 512 megatavua RAM-muistia, 10 gigatavun tallennustila ohutprovisio-oinnilla (sallii levytilan laajentumisen tarvittaessa) ja asennettavaksi käyttöjärjestelmäksi valitaan Debian 5 (64-bit). Viimeisin kohta

ei täysin vastaa todellisuutta asennettavan käyttöjärjestelmän ollessa Debian 6, mutta käytännössä ongelmia ei yhteensopivuuden vuoksi tule. Tallennustila allokoidaan fyysisesti levypalvelimelle.

Varsinainen asennus alkoi virtuaalisen koneen käynnistyksellä. Aluksi tuli vain ilmoitus käynnistyslevyn uupumisesta, joka on asiaankuuluvaa käyttöjärjestelmän vielä puuttuessa. Asennusmediana käytetään aiemmin luokan työasemalle Internetistä ladattua Debian-asennusmediaa, joka on muodoltaan levykuva ISO. Tämä levykuva on mahdollista tallentaa optiselle medialle, kuten DVD:lle, tai sen voi sellaisenaan näyttää virtuaalikoneelle. Levykuva saadaan näkyviin virtuaalikoneelle valikosta. Kun virtuaalikone sen jälkeen käynnistetään uudelleen, huomaa se sille annetun asennusmedian ja sieltä tarvitun käynnistyslevyn. Käyttöjärjestelmän asennus käynnistyy ja ruutuun tulee kuvion 2 mukainen näkymä.

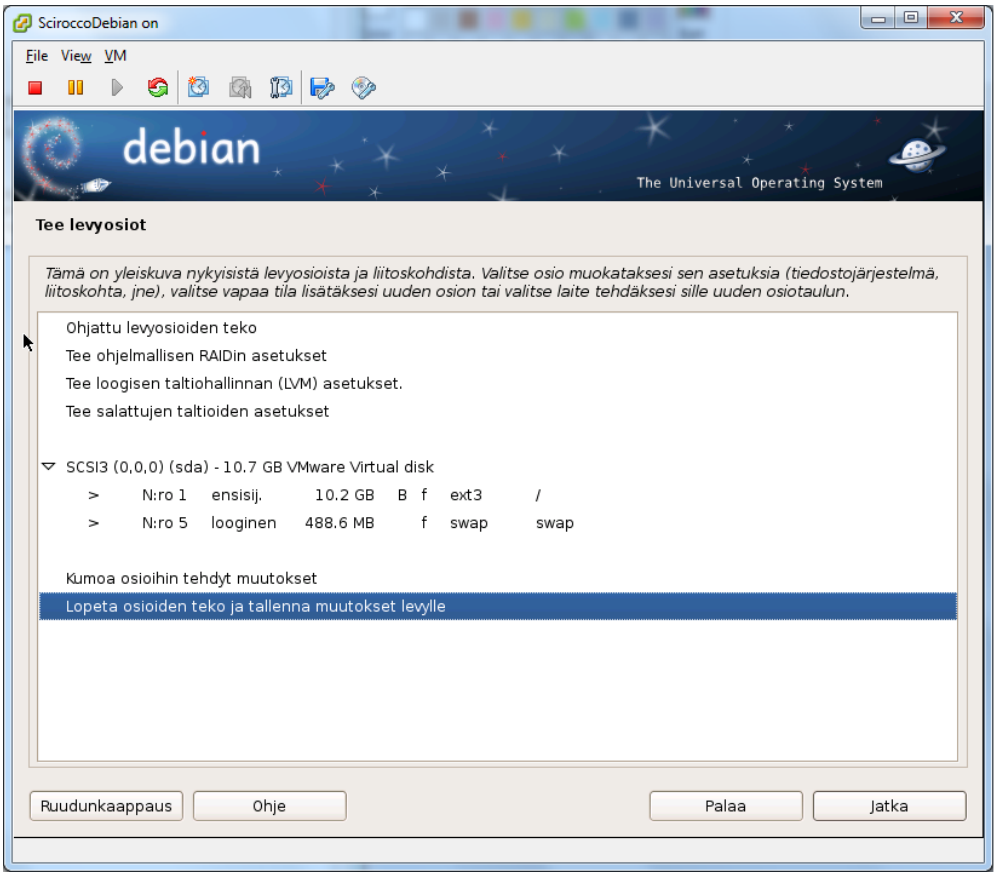


Kuvio 2. Debianin asennusohjelman aloitusruutu.

Vaikka graafista käyttöliittymää ei halutakaan asentaa, valitaan asennustavaksi silti graafinen asennus mahdollisten virheiden minimoimiseksi. Kieliasetuksiksi asennetaan suomi. Palvelimen nimeksi valitaan Vomus. Sana viittaa Windowsin puolella käytettyyn asennusohjelmaan Install Shieldiin, mutta erinäisistä syistä sana valitaan myös tähän tarkoitukseen. Verkkoalueeksi asetetaan kjlabra.kajak.fi, mutta tämä voidaan tarvittaessa vielä jälkikäteen vaihtaa.

Linux-ympäristössä käyttäjätunnuksilla ja niiden oikeuksilla on normaalia Windows-käyttöä suurempi merkitys. Kun tavallisesti Windowsia käytetään kotikäytössä yleensä täysillä järjestelmänvalvojan oikeuksilla, ei Linuxissa oletuksena tämä ole sallittua. Tästä syystä jo asennuksessa luodaan kaksi käyttäjätunnusta: pääkäyttäjä, jolla on käytännössä suurimmat mahdolliset oikeudet suorittaa asioita, sekä peruskäyttäjä, jolla on tarkoitus suorittaa suurin osa normaaleista toimenpiteistä. Tarvittaessa käyttäjän oikeuksia korotetaan esimerkiksi ohjelmien asennusta varten. Kaikki tunnukset suojataan salasanoilla.

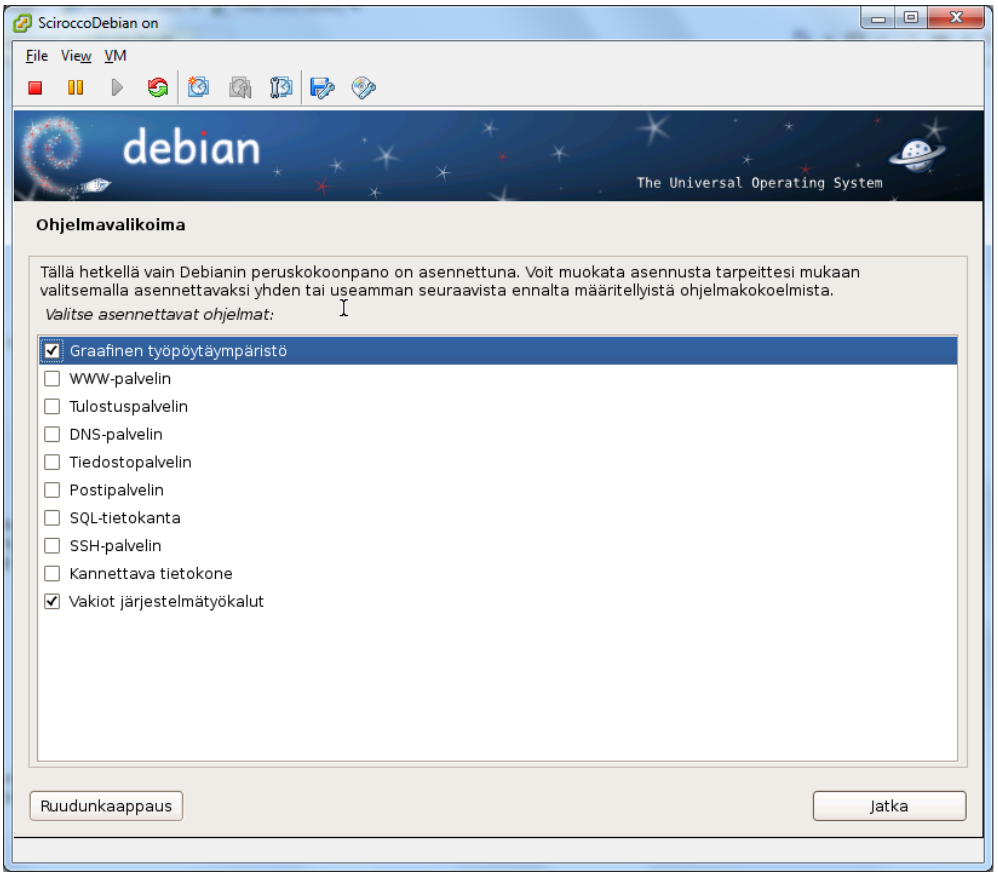
Levyjärjestelmään luodaan omat osiot kaikille oleellisille kansioille osiontityökalua. Näitä ovat /home, /usr, /var ja /tmp, jotka näkyvät kuviossa 3. Toimenpide ei ole välttämätön ja yleensä sitä suositellaankin lähinnä suurikokoisille levyjärjestelmille, mutta tulevaisuutta ajatellen toimenpiteestä ei pitäisi tulla ongelmia. (Garner 2012.)



Kuvio 3. Osiointityökalu.

Haluttaessa jo asennusvaiheessa voidaan lisätä useita asennuslähteitä. Tällöin uusia ohjelmistoja voidaan asentaa jo asennusvaiheessa, mutta tässä projektissa se ei ole tarpeen. Asennuspalvelimeksi päivityksiä varten jätetään oletuksena oleva ftp.fi.debian.org. Välipalvelinta ei myöskään lisätä asennukseen.

Asennuksen valmistuttua voidaan valita kuvion 4 mukaisesta listasta ohjelmavalikoimat. Valittavana ovat esimerkiksi www-palvelin, jonka valitsemalla järjestelmä asentaa roolin vaatimat sovellukset. Koska tarkoituksena on itse asentaa tarvittavat ohjelmistot, valitaan asennettavaksi ainoastaan vakiot järjestelmätyökalut.



Kuvio 4. Ohjelmavalikoima.

Järjestelmä asentaa vielä koneen käynnistyksessä vaaditun käynnistyslataaja GRUB:n jonka jälkeen käyttöjärjestelmä on käyttövalmis. Ennen uusien ohjelmistojen asennusta tehdään varmuuden vuoksi sovellusten päivitys, jotta mahdolliset tietoturvapäivitykset ovat asennettuna. Päivitys tapahtuu kahdessa vaiheessa: ensin päivitetään päivitystietokanta apt-get update-komennolla ja sen jälkeen itse sovellukset apt-get upgrade-komennolla. Kumpikin suoritetaan järjestelmänvalvojan oikeuksilla lisäämällä eteen sudo ja antamalla sitten salasana pyydettäessä.

4.2 Tarvittavien ohjelmistojen asennus

Palvelimelle on tarkoitus asentaa aiemmin mainitut LAMP, Pear ja GD. Ohjelmat ovat saatavilla Debianin omasta ohjelmalähteestä, joten asennukseen riittävät oikeat asennuskomennot sekä verkkoyhteys. Ohjelmien asennus suoritetaan sudo-komennon kanssa:

- LAMP: apt-get install apache2 php5 apache2.2-common libapache2-mod-auth-mysql php5-mysql mysql-server

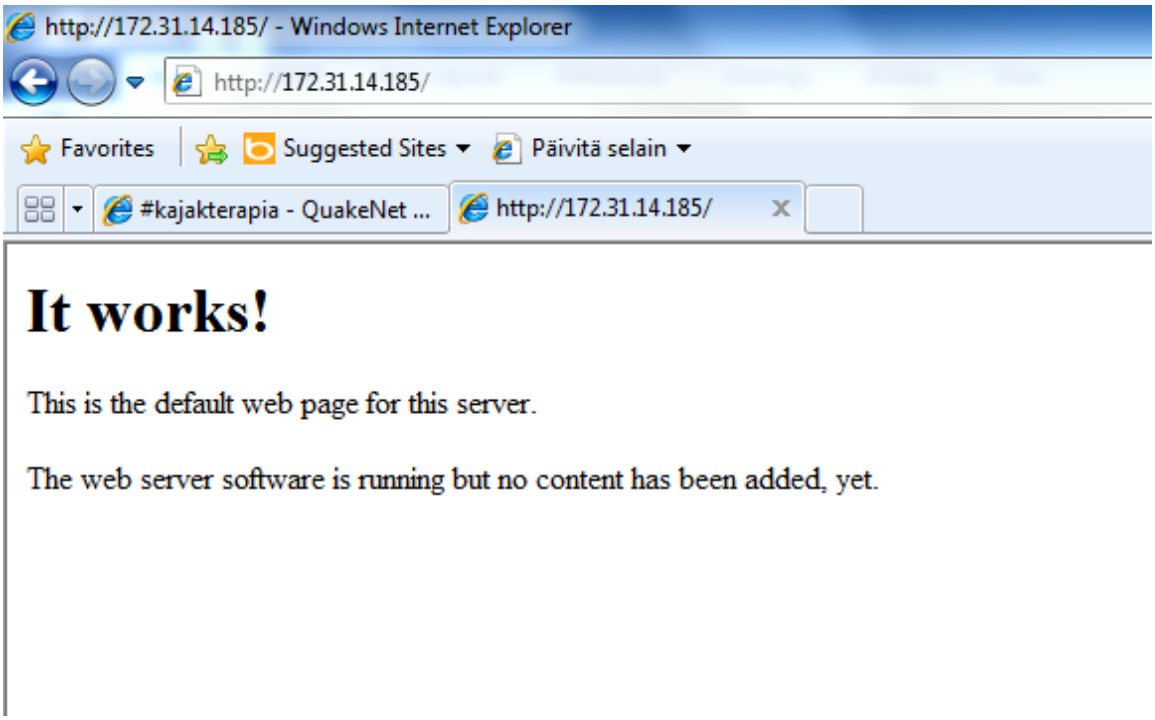
Kuten komennosta voi huomata, LAMP on itseasiassa kokoelma useaa eri ohjelmistoa: Apache 2, PHP5 sekä MySQL. Yhdessä nämä muodostavat palvelukokonaisuuden.

Apache2 on Internetin HTTP-protokollan käyttöön tarkoitettu ohjelmisto. Sen avulla voidaan luoda Internet-sivustoja ja se tarjoaa turvallisen alustan sivustoille. Palvelimella Apache tarvitaan jotta ajoneuvo voi saada palvelimeen yhteyden sekä HTTP- että suojatulla HTTPS-yhteydellä. (Apache HTTP server project, 2012.)

PHP on ohjelmointikieli jota käytetään yleisesti web-sisältöjen luonnissa. Se on niin sanottu skriptikieli, eli koodi tulkitaan vasta suoritusvaiheessa. Tämä mahdollistaa sen, että komentoja voidaan automatisoida ilman varsinaisia muita ohjelmakieliä. (The PHP Group 2012.)

MySQL on tietokantasovellus jolla voi luoda ja hallita tietokantoja. Palvelimessa tietokanta säilyttää siihen siirretyt tiedot ja mahdollistavat tietojen jatkokäytön. Tietokantoja käytetään laajasti eri palveluissa ja Internet-sivustoissa. (Oracle 2012.)

Asennuksen onnistumisen voi tarkistaa käyttämällä Internet-selainta ja menemällä sillä palvelimen IP-osoitteeseen. Tällöin näkyviin pitäisi tulla etusivu, joka sijaitsee palvelimella /var/www/-kansiossa ja oletuksena palvelin näyttää sieltä etusivuna käytetyn index.html-tiedoston sisällön. Tässä tapauksessa sivusto toimii ja ruutuun ilmestyy kuvion 5 mukainen teksti ”It works!”. Mahdollista tarvetta varten sallitaan myös paikallinen sivutarkkailu muokkaamalla Apache2:n porttiasetuksia komennolla ”nano /etc/apache2/ports.conf” ja vaihtamalla sinne ”Listen 80” muotoon ”Listen 127.0.0.1:80”.



Kuvio 5. Sivusto toimii. Osoiterivillä oleva teksti ”http://” ilmaisee sivuston olevan salaamaton.

Seuraavana vaiheena on asentaa dynaamisten kuvien tuki GD. Se tapahtuu pääkäyttäjänä komennolla ”apt-get install php5-gd”.

Tässä vaiheessa asennetaan phpMyAdmin. Sen avulla voi Internet-selaimen kautta hallita tietokantoja ja se helpottaa myöhemmin vanhan palvelimen tietokannan siirtämistä uudelle alustalle. Asennus tapahtuu komennolla ”apt-get install phpmyadmin”.

Apachen ja MySQL:n uudellenkäynnistyksen jälkeen asennetaan vielä viimeinen osa, eli Pear. Asennus tehdään komennolla ” apt-get install php-pear”. Asennus ei ole monimutkaista ja se tapahtuu ilman merkittäviä huomioita. Lopulta kone käynnistetään vielä kerran uudelleen

Tarvittavien ohjelmistojen jälkeen on aika keskittyä tietoturvaan. Palvelimen ollessa rungoltaan HTTP-palvelin on suurin tietoturvauhka tällöin Internetistä tulevat mahdolliset hyökkäykset ja siksi palomuurilla on merkittävä asema turvallisuuden ylläpidossa. Palomuuri tehdään manuaalisesti kirjoittamalla tarvittavat komennot tiedostoon.

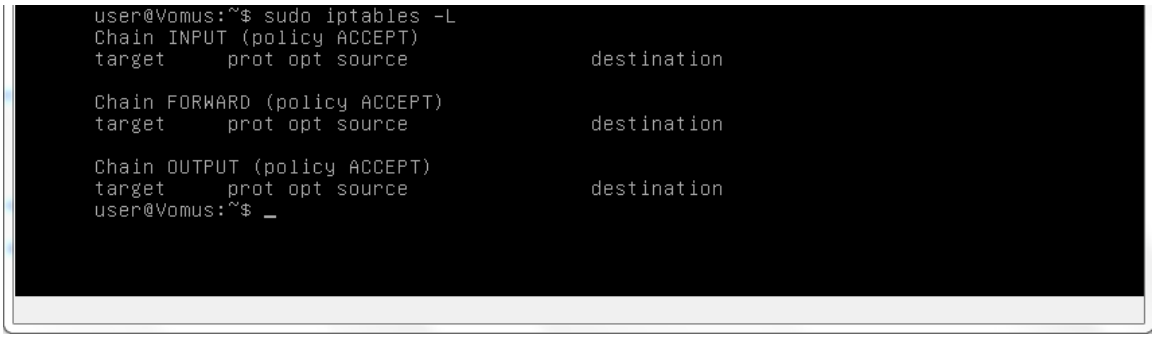
4.3 Palomuuri

Palomuuri on järjestelmä, joka suodattaa verkkoyhteyttä. Sen tarkoituksena on estää ei-haluttuja yhteyksiä pääsemästä määränpäähänsä ja samalla se torjuu mahdolliset hyökkäysyritykset. Rakennettava palvelin toimii tuotantokäytössä verkon DMZ-alueella, jolloin palvelimelle tulee tehdä oma palomuuri.

Verkkoyhteydet toimivat porttien kautta. Kun yhteys luodaan, ohjelma lähettää yhteyspyynnön halutun osoitteen porttiin ja riippuen vastaanottajan palomuurista ja ohjelmistojen asetuksista yhteys vastaanotetaan ja siihen vastataan. Porttien avoimuutta säätämällä voidaan luoda kestävä ja turvallinen palomuuri. (vlaurie.com 2012.)

Portit voivat olla väliltä 0-65535 ja niitä käytetään pääasiallisesti UDP- ja TCP-protokollien yhteydessä. Portit ovat suurelta osin standardisoitu: portit väliltä 0-1023 ovat nimeltään tunnettuja portteja, johon kuuluvat yleisten yhteysprotokollien portit; 1024–49151 ovat rekisteröityjä portteja, joita voi ostaa ja joiden rekisteröintiä ja tietokantaa ylläpitää kansainvälinen Internet Assigned Numbers Authority (IANA) sekä portit 49152–65535, jotka ovat dynaamisia ja yksityisiä portteja. (Abrams 2004.)

Palomuuria tehdessä pääajatuksena on, että vain tarvittavat portit pidetään avoinna ja kaikki muu suljetaan. Tämä tuo haastetta, sillä käytännössä kaikki verkkoyhteydet ensin suljetaan ja sitten yksi kerrallaan portteja avataan ja testataan. Jos konfigurointi tehtäisiin toisinpäin avoimesta suljetuksi, ei voitaisi olla varmoja siitä, että järjestelmään ei jää tukkimattomia aukkoja. Kuviossa 6 on Debianin palomuurisäännöt oletusasetuksissaan ilman muokkauksia. Palomuurin säännöt näkee Debianissa komennolla ”iptables –L”.

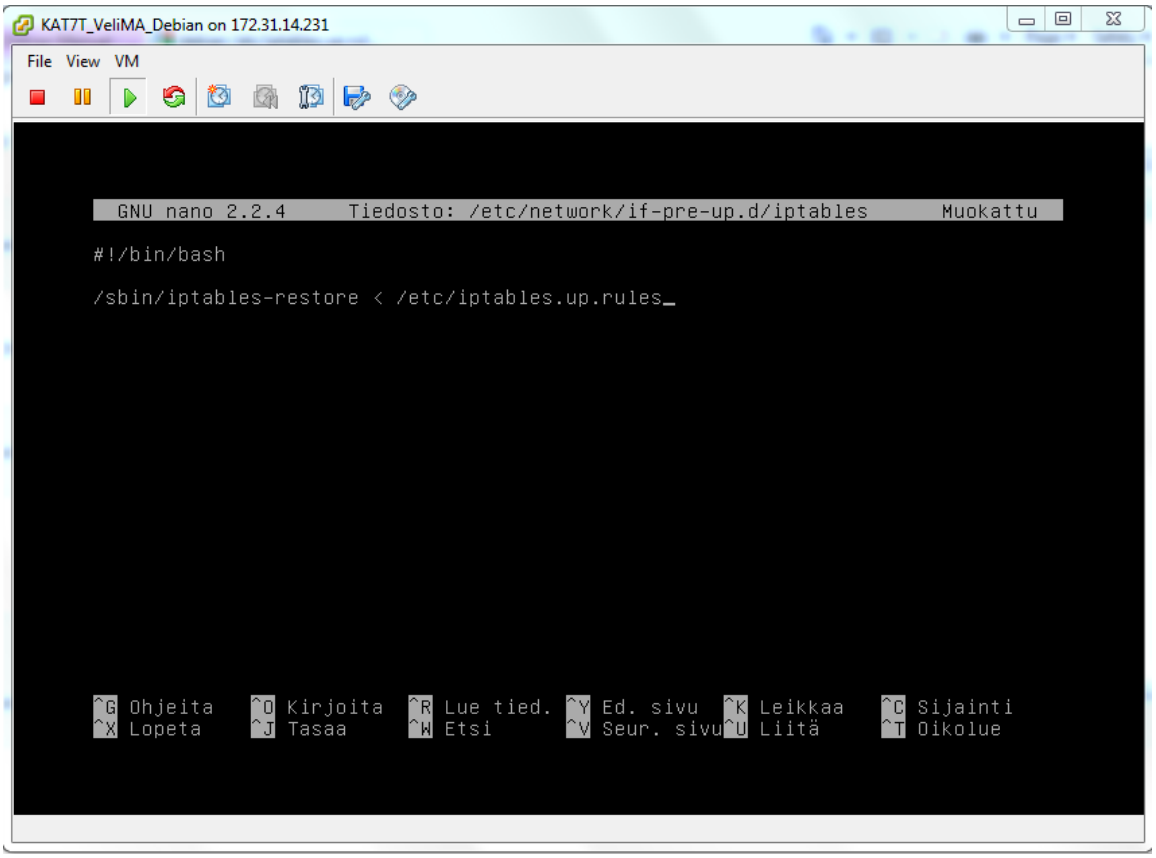


Kuvio 6. Debianin palomuurisäännöt oletusasetuksilla. Missään kolmesta verkkosuunnasta ei ole rajoituksia.

4.3.1 Palomuurin konfigurointi

Ennen aloitusta kartoitetaan mitä portteja halutaan avoimeksi. Tässä tapauksessa avoimeksi halutaan HTTP-, HTTPS-, DNS-, FTP-, NFS- ja ping-portit. HTTP- ja HTTPS-portit avataan, koska tutkimusauton ja palvelimen välinen tiedonsiirto toteutetaan www-sivuston kautta ja siten käyttäen HTTP-portteja 80 ja 8080 sekä suojattua HTTPS-porttia 443. FTP-portit 20, 21, 1023 sekä 1024 avataan päivitysten turvaamiseksi ja samasta syystä myös lähtevä DNS-portti 53 avataan. NFS-portit 111, 2049, 32764-32769 sekä 52049-52050 avataan järjestelmän sisäisestä vaatimuksesta. Testaamista helpottamaan avataan myös PING-komennon käyttämä portti 8. Kaikki muut portit suljetaan.

Palomuuri rakennetaan luomalla tiedosto, jolla korvataan järjestelmän palomuuritietokanta. Tässä tapauksessa tiedostona johon palomuurisäännöt tehdään on /etc/iptables.rules ja sitä muokataan esimerkiksi nano-komennolla. Säännöt aktivoidaan root-käyttäjänä komennolla iptables-restore < /etc/iptables.rules ja otetaan käyttöön iptables-save > /etc/iptables.up.rules. Lopulta säännöt saadaan aktivoitua tekemällä skriptin sijaintiin /etc/network/if-pre-up.d/iptables ja vaihtamalla sen oikeudet siten että sen voi suorittaa. Skriptin sisältö näkyy kuviossa 7. Lopulta käyttöjärjestelmän uudelleenkäynnistys ottaa palomuurisäännöt käyttöön.



Kuvio 7. Uudet palomuurisäännöt käyttöön ottava skripti.

Palomuuria tehdessä määritellään porttikohtaisesti tietoliikenneprotokolla, onko yhteys saapuva (input), lähtevä (output) vai välittävä (forward), sekä onko portti kohteena (--dport) vai lähteenä (--sport). Esimerkiksi HTTP-portin avaava komento ”-A INPUT -p tcp --dport 80 -j ACCEPT” ilmaisee, että TCP-protokollalla saapuva yhteys kohteenaan portti 80 hyväksytään läpi. Komennon lopussa ilmaistaan yhteydelle haluttu toimenpide: salli (accept) tai estä (drop ja reject). Dropin ja rejectin ero on siinä, että drop-komentoa käytettäessä toinen osapuoli ei saa tietoa yhteyden estämisestä eikä myöskään saa varmistusta kohteen olemassa olosta, vaan yhteys katkeaa ajallaan itsestään vastauksen uupuessa. Reject-komennolla toinen osapuoli saa tiedon että yhteys estettiin, mutta samalla se myös huomaa, että kohdeosoitteessa ja portissa todellakin on jotain. Tietoturvan kannalta tässä palvelimessa käytetäänkin drop-komentoa, sillä palvelin on tarkoitettu yksityiskäyttöön.

Alun perin, kuten kuviossa 8 näkyy, oli tarkoitus jättää FTP- , DNS-, ja NFS-portit suljetuiksi, mutta niiden määrittämättä jättäminen aiheutti ongelmia järjestelmän ohjelmistojen päivityksessä ja järjestelmän sisäisen sähköpostiohjelman toiminnassa. Päivitykset ovat tärkeydeltään niin suuria, että niiden toimivuus on välttämätöntä ja sähköpostipalvelin tarvitaan jos halutaan, että järjestelmä pystyy varoittamaan tarvittaessa ongelmista.

```
*filter

# Allow HTTP and HTTPS
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 8080 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT

# Allow ping
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

# Reject all other inbound unless explicitly allowed earlier
-A INPUT -j REJECT
-A FORWARD -j REJECT

COMMIT
```

Kuvio 8. Ensimmäinen palomuurikokeilu, jossa on sallittuna vain sisäänpäin tulevat HTTP-, HTTPS- ja PING-protokollat.

Palomuurin asetusten lopussa määritellään kaikki muut yhteydet suljetuiksi. Määritys suoritetaan komennoilla:

- -A INPUT -j DROP
- -A FORWARD -j DROP

Huomionarvoisesti output-jätetään estämättä, sillä palvelimelta lähtevät yhteydet eivät oletettavasti ole tietoturvauhaltaan suuria. Lisäksi kaikkien lähtevien yhteyksien määritys lisäisi palomuurin muokkaustarvetta reippaasti, sillä samalla kaikki erilliset yhteydet pitäisi portti kerrallaan tutkia ja sallia, joka se veisi runsaasti aikaa.

Palomuurin konfigurointi on kohtuullisen monimutkaista. Asiat tulee kyetä tajuamaan monisuuntaisesti yhteyksien toimiessa useaan eri suuntaa usealla eri tasolla. Tästä johtuen asiat on mahdollista saada hyvinkin sekaisin ja tällöin virheiden löytäminen määrityksistä voi olla todella vaikeaa.

Päivitysten toimimattomuus ja yksittäisten palvelujen (mm. NFS common utilities) virheilmoitukset pakottavat testaamaan tarvittavien porttien avaamista. Käytännössä tämä tapahtuu etsimällä Internetistä tarvittavat portit ja avaamalla ne palomuurisäännöissä. Ongelmaksi muodostuukin tajuta yhteyksien suunta, sillä esimerkiksi aiemmin mainittu NFS ei saanut yhteyttä tällä samalla palvelimella olevaan toiseen palveluun, eli yhteys ei käytännössä edes kohdistunut Internetiin. Jos logiikkaa ei ymmärrä, menee konfigurointi helposti hakuammunnaksi ja palomuurin suojauskyky heikkenee merkittävästi mahdollisten virheiden vuoksi.

Kuvioissa 9, 10, ja 11 näkyy valmiin palomuurin konfiguraatio. Osa komennoista on monipuolisemmassa muodossa kirjoittamisen säästämiseksi ja osassa komennoista porttiväli määritellään yhdessä isossa alueessa. Kuviossa 9 huomioitavaa ovat myös viisi ensimmäistä riviä, joissa sallitaan yhteyksien muodostus käyttäen fyysistä, johdolla tehtyä yhteyttä laitteiston tietystä liittimestä. Nämä säännöt eivät itsestään mahdollista yhteyttä, mutta niiden avulla voidaan hallita yhteyteen käytettyä liitäntää.

```
  GNU nano 2.2.4         Tiedosto: /etc/iptables.rules

*filter
:ETH0_IN - [0:0]
-A INPUT -i lo -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A INPUT -i eth0 -j ETH0_IN

# Allow HTTP and HTTPS
-A INPUT -p tcp --sport 80 -j ACCEPT
-A INPUT -p tcp --dport 80 -j ACCEPT
-A INPUT -p tcp --dport 8080 -j ACCEPT
-A INPUT -p tcp --dport 443 -j ACCEPT

# Allow ports for DNS lookup
-A INPUT -p udp --sport 53 --dport 1024:65535 -j ACCEPT

# Allow FTP for updates
-A INPUT -p tcp --sport 20 -j ACCEPT
-A OUTPUT -p tcp --dport 20 -j ACCEPT
-A OUTPUT -p tcp --dport 21 -j ACCEPT
-A INPUT -p tcp --sport 21 -j ACCEPT
```

Kuvio 9. HTTP-, HTTPS-, DNS- ja FTP-protokollat sallivat säännöt.

```
-A OUTPUT -p tcp --dport 1023 -j ACCEPT
-A INPUT -p tcp --sport 1023 -j ACCEPT
-A INPUT -p tcp --sport 1024 -j ACCEPT
-A OUTPUT -p tcp --sport 1024 -j ACCEPT

# Allow NFS mounts on local network
-A ETH0_IN -p tcp -m state --state NEW -m tcp --dport 111 --tcp-flags SYN,RST,A$
-A ETH0_IN -p udp -m state --state NEW -m udp --dport 111 -j ACCEPT
-A ETH0_IN -p udp -m state --state NEW -m udp --dport 2049 -j ACCEPT
-A ETH0_IN -p tcp -m state --state NEW -m tcp --dport 2049 -j ACCEPT
-A ETH0_IN -p tcp -m state --state NEW -m tcp --dport 32764:32769 -j ACCEPT
```

Kuvio 10. Loput FTP- ja NFS-protokollan säännöt.

```
-A ETH0_IN -p udp -m state --state NEW -m udp --dport 32764:32769 -j ACCEPT
-A ETH0_IN -p udp -m state --state NEW -m udp --dport 52049 -j ACCEPT
-A ETH0_IN -p udp -m state --state NEW -m udp --dport 52050 -j ACCEPT


# Allow ping
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT

# Drop all other inbound unless explicitly allowed earlier
-A INPUT -j DROP
-A FORWARD -j DROP
```

Kuvio 11. Loput NFS-protokollan säännöt ja PING-komennon mahdollistava sääntö.

5 YHTEYDEN SALAUS

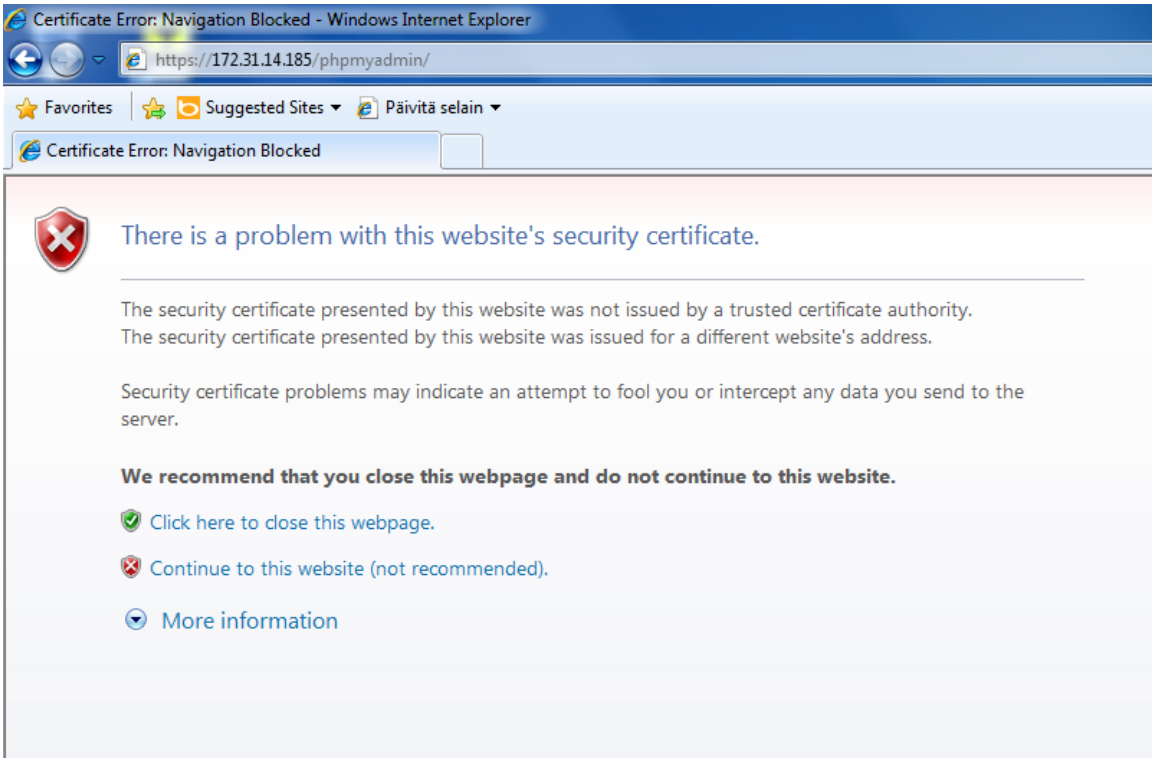
Tavallinen HTTP-protokolla on salaamatonta. Se tarkoittaa, että liikkuva data on selväkielistä ja kaapatessa se on helposti sivullisten luettavissa. Tätä tietoturvauhkaa voidaan ehkäistä salaamalla yhteys, jolloin vain lähettäjä ja vastaanottaja voivat lukea liikkuvaa dataa. HTTP-yhteyden salattu muoto on HTTPS ja sitä käytetään yleisesti Internetissä turvallisuutta vaativilla sivustoilla. (Wikipedia 2012.)

Salattu yhteys voidaan toteuttaa kahdella eri salausmetodilla, symmetrisellä tai epäsymmetrisellä, sekä näiden yhdistelmillä. Symmetrinen ja epäsymmetrinen salaus eroavat toisistaan salaukseen käytettävien avainten määrissä ja käyttötavoissa. Symmetrisessä salauksessa data salataan ja avataan samalla salausavaimella ja tämä vaatiikin että avain on erikseen toimitettu datan vastaanottajalle. Epäsymmetrisessä salauksessa data salataan ja avataan eri avaimilla, jolloin salaukseen käytetään vastaanottajan yleisesti jakamaa julkista salausavainta, mutta salauksen saa auki vain vastaanottajan omassa tiedossa olevalla salaisella avaimella. Epäsymmetrinen salaus on tehokas, mutta samalla raskas salaus, ja tästä syystä käytännössä käytetään kummankin salausmetodin sekoitusta. Tällöin prosessi toimii kuten epäsymmetrisessä salauksessa, mutta alussa toiselle osapuolelle toimitetaan kertakäyttöinen salausavain, jolla voi purkaa viestin. Kertakäyttöisyys mahdollistaa sen, että avain voi olla lyhyempi ja siten helpompi murtaa, mutta koska avainta ei käytetä kuin kerran ei sen kaappaamisesta hyödy mitenkään. (Heinonen 2002.)

Rakennettavan palvelimen salauksena käytetään SSL-salausta. Tällöin asiakaskoneen ottaessa yhteyden se varmistaa palvelimen luotettavuuden ja sen jälkeen lähettää palvelimen julkisella avaimella salatun kertakäyttöisen avaimen palvelimelle. Tällä avaimella salataan kaikki sillä istunnolla tapahtuva dataliikenne. Palvelimen luotettavuus todennetaan varmenteilla, jotka voi joko luoda itse tai ostaa kaupallisilta toimijoilta. (Symantec 2012.)

Salattua yhteyttä varten palvelin tarvitsee kaksi salausavainta: julkisen ja salaisen. Nämä luodaan palvelimelle käyttäen varmennetta, jolla palvelin esittäytyy siihen yhteyttä ottavalle asiakkaalle. Varmenteen saamiseksi on olemassa kolme eri vaihtoehtoa: Voi ostaa luotetun varmenteen kaupalliselta ja tunnetulta alan toimijalta, rakentaa sen itse tai antaa palvelimen itse luoda allekirjoittamansa varmenne. Näistä ostaminen olisi ihanteellisin ratkaisu, mutta

kaupallisten varmenteiden kova hinta tekee siitä epärealistisen. Itse tehty ja allekirjoitettu varmenne aiheuttaa sen, että asiakkaan ottaessa yhteyden palvelimelle saa se kuvion 12 mukaisen varoituksen mahdollisesti vaarallisesta sivustosta.



Kuvio 12. Selaimen antama varoitus varmenteen epäluotettavuudesta.

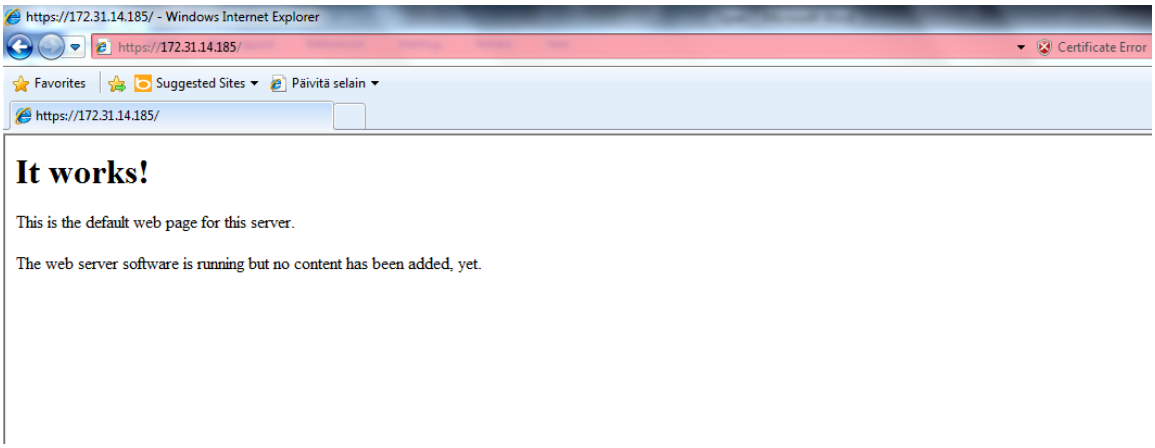
5.1 Varmenteen luonti

Varmenne voidaan tehdä itse kahdella eri tavalla: helposti antaen palvelimen itse luoda se, tai vaikeasti tekemällä se käsin itse. Itse luotu varmenne sallisi suuremmat muokkausmahdollisuudet varmenteen sisältämiin tietoihin, mutta käytännössä se vaatii myös runsaasti sivustoasetusten muokkaamista järjestelmässä, joten on helpompaa antaa palvelimen itse luoda tarvitsemansa varmenne.

HTTPS-sivuston käyttöönotto ja varmenteen luonti tapahtuu kahdella komennolla root-käyttäjänä:

- a2enmod ssl
- a2ensite default-ssl

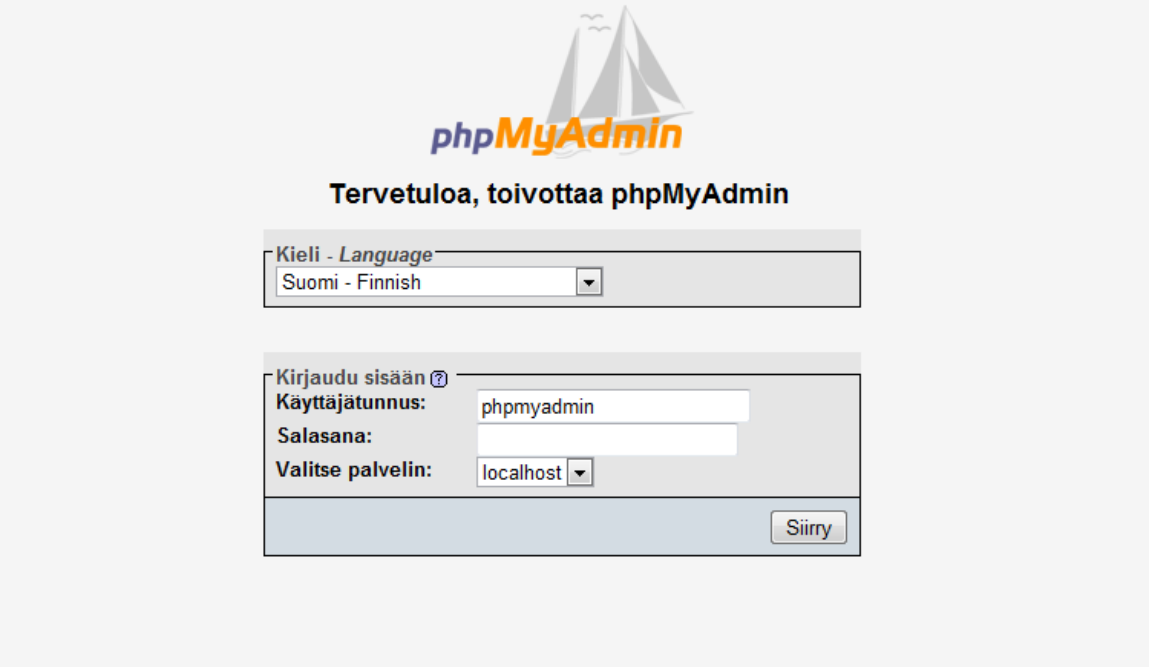
Ensin mainittu mahdollistaa SSL-salauksen salauksen käytön ja luo samalla tarvittavat avaimet ja varmenteen. Toinen komento ottaa salauksen käyttöön sivustossa. Salaus aktivoituu käynnistämällä apache uudelleen komennolla ”/etc/init.d/apache2 restart”. Salauksen voi todeta kuviosta 13 osoiterivillä olevasta ”https://”-tekstistä ja huomionarvoista on myös osoiteriviltä löytyvä sertifikaattivaroitus, joka johtuu varmenteesta.



Kuvio 13. Se toimii taas.

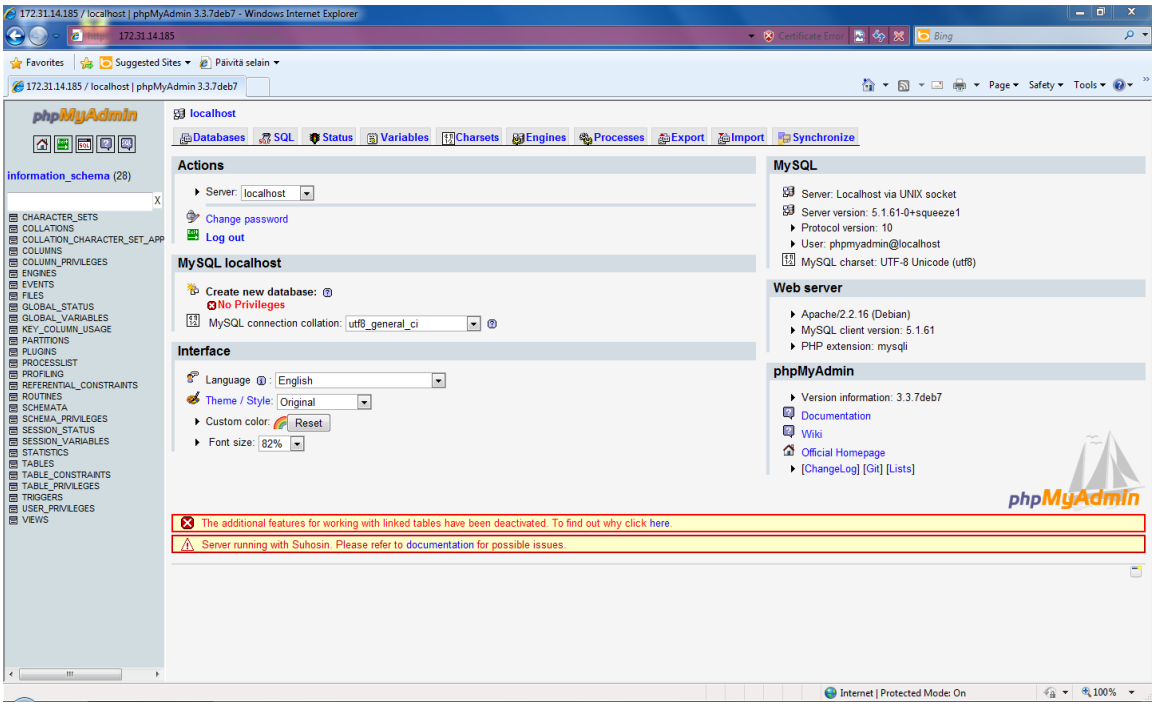
6 PHP JA TIETOKANTA

Tutkimusautoa varten palvelimelle luodaan tietokanta, johon ajoneuvon lähettämä tieto tallennetaan ja johon siirretään testipalvelimessa oleva tietokanta. Tietokantaa varten asennettiin jo aiemmassa vaiheessa phpMyAdmin-ohjelmisto, jonka avulla tietokantoja voidaan hallita Internet-selaimen avulla. Ohjelmistoa voi käyttää lisäämällä IP-osoitteen perään ”/phpmyadmin”. Kuviossa 14 on phpMyAdminin kirjautumissivusto.



Kuvio 14. phpMyAdminin kirjautumissivu.

Ensimmäisellä kirjautumiskerralla palveluun pääsee vain root-käyttäjänä, jolloin salasanana toimii MySQL-asennuksessa annettu salasana. Heti ensimmäisen kirjautumisen jälkeen luodaan uusi rajoitetumpi käyttäjätunnus, sillä root on liian jykevä käyttäjä käyttöön. Lisäksi pakotetaan hallintasivusto toimimaan salattuna muokkaamalla ”/etc/phpmyadmin/config.inc.php”-tiedostoa lisäämällä sinne ” $cfg['ForceSSL'] = 'true';”. Kuviossa 15 näkyy phpMyAdminin käyttöliittymä rajoitetummalla käyttäjällä.



Kuvio 15. phpMyAdminin selainkäyttöliittymä.

Vanhasta palvelinalustasta siirretään valmis www-sivusto sekä tietokannat uuteen alustaan. Ne siirretään pakattuina muistitikkua käyttäen etäyhteyden muodostamiseen pystyvälle työasemalle, josta ne myöhemmin siirretään palvelimelle. Tiedostojen siirto ei onnistu suoraan VMwaren vSpherellä, joten käytettäväksi valitaan yleisesti käytetty tiedonsiirtotapa SSH.

SSH:n käyttöönotto vaatii, että palvelimelle asennetaan yhteyttä varten sovellus ja että palomuuriin sallitaan sen vaatima portti. Ohjelman asennus suoritetaan samalla tavoin kuin muiden ohjelmien asennus:

- apt-get install ssh

Asennuksen jälkeen sallitaan palomuuriin SSH-protokollan mukainen portti 22:

- -A INPUT -p tcp --dport 22 -j ACCEPT

Palomuurin konfiguroinnin jälkeen palvelin on valmis vastaanottamaan tiedostoja. Tiedostojen siirto vaatii työasemalle SSH-asiakasohjelmiston, jonka avulla tiedostoja voi siirtää. Ohjelmana käytetään PSCP-ohjelmaa. Tässä vaiheessa siirretään ainoastaan www-sivusto, sillä tietokanta siirretään myöhemmin myPHPAdminilla. Sivusto siirretään oletuskansioon jolloin se korvaa tämän hetkisen sivuston.

Ohjelmalla otetaan yhteys palvelimen sen hetkiseen IP-osoitteeseen. Tämä tapahtuu käyttämällä työaseman komentoriviä, tässä tapauksessa Windowsin command promptia. Komentoriville kirjoitetaan seuraava komento:

- pscp –P 22 –r c:Users\kjlabra\Desktop\sivut* root@172.31.14.185:/var/www/

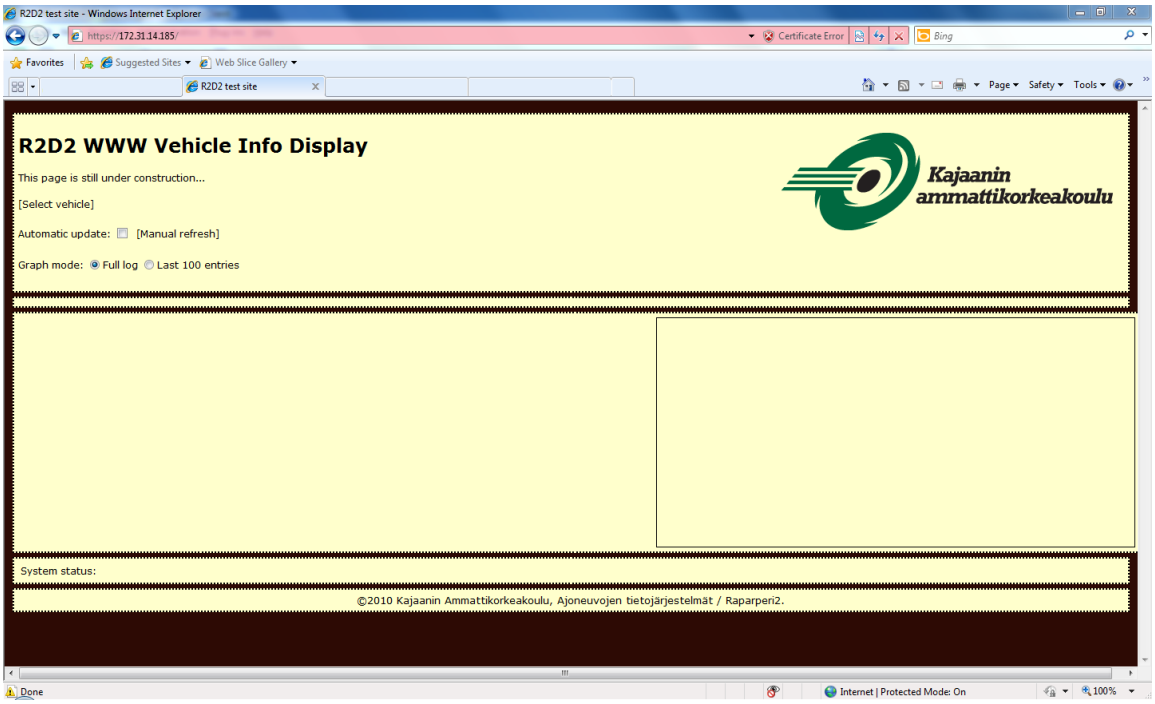
Komento tarkoittaa, että PSCP-ohjelmalla siirretään porttia 22 käyttäen c-asemalla olevasta kansiosta kaikki tiedostot IP-osoitteen osoittaman palvelimen /var/www/-kansioon käyttäen root-tunnusta. Palvelin vaatii ennen siirtoa root-käyttäjän salasanan ja sen jälkeen siirto alkaa. Siirto näkyy kuviosta 16. IP-osoite ei ole palvelimen lopullinen, joten se vaihdetaan palvelimen valmistuttua.

```
D:\ssh>pscp -P 22 -r c:Users\kjlabra\Desktop\sivut\* root@172.31.14.185:/var/www
/
root@172.31.14.185's password:
graphdraw.php              |  0 kB |   1.0 kB/s | ETA: 00:00:00 | 100%
kamk_logo.png              |  7 kB |   7.7 kB/s | ETA: 00:00:00 | 100%
config.php                 |  0 kB |   0.4 kB/s | ETA: 00:00:00 | 100%
graphdraw.php              |  0 kB |   1.0 kB/s | ETA: 00:00:00 | 100%
imgfunc.php                |  1 kB |   1.1 kB/s | ETA: 00:00:00 | 100%
reqs.php                   | 14 kB |  14.2 kB/s | ETA: 00:00:00 | 100%
index.php                  |  2 kB |   2.0 kB/s | ETA: 00:00:00 | 100%
jquery.min.js              | 70 kB |  70.6 kB/s | ETA: 00:00:00 | 100%
r2d2.js                    |  5 kB |   5.1 kB/s | ETA: 00:00:00 | 100%
r2d2_development_v001.rar  | 21 kB |  21.9 kB/s | ETA: 00:00:00 | 100%
r2d2_development_v003.rar  | 24 kB |  24.0 kB/s | ETA: 00:00:00 | 100%
r2d2_development_v004.rar  | 33 kB |  33.9 kB/s | ETA: 00:00:00 | 100%
reqhandler.php             |  2 kB |   3.0 kB/s | ETA: 00:00:00 | 100%
styles.css                 |  3 kB |   3.2 kB/s | ETA: 00:00:00 | 100%
_index.html                |  2 kB |   2.1 kB/s | ETA: 00:00:00 | 100%

D:\ssh>
```

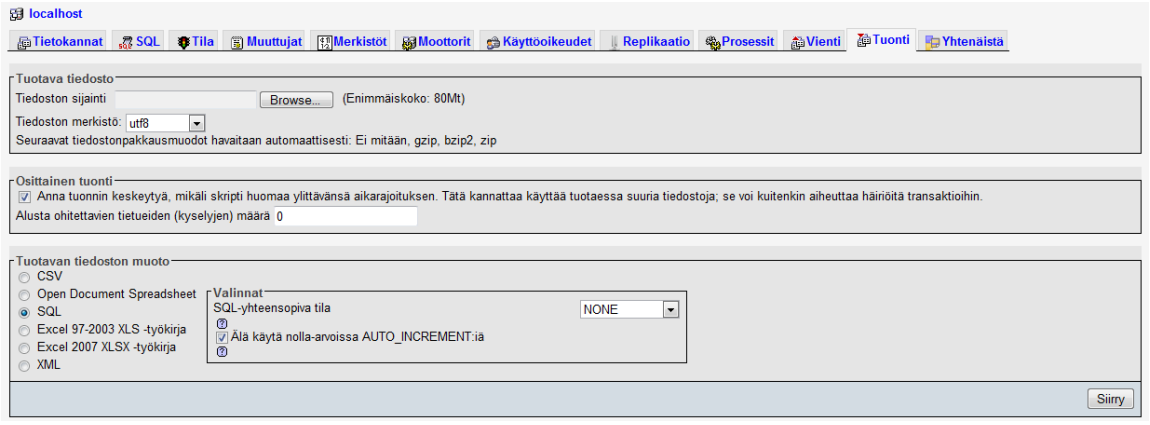
Kuvio 16. Tiedostot ovat siirtyneet.

Siirron jälkeen palvelin käynnistetään uudelleen. Tämän jälkeen selaimella testataan uuden sivuston toimivuus, jolloin näkyviin tulee kuvion 17 mukainen näkymä.



Kuvio 17. Uusi sivusto paikallaan.

Sivuston toimivuuden varmistuttua siirretään vanhan palvelimen tietokanta rakennettuun uuteen palvelimeen. Tietokanta on työasemalla ”.sql”-tiedostona, eli se sisältää tietokannan ja siihen liittyvät käyttäjätunnukset kokonaisuudessaan. Siirto tapahtuu automaattisesti phpMyAdminista löytyvällä graafisella tuontityökalulla, joka näkyy kuviossa 18.



Kuvio 18. PhpMyAdminin tietokantojen tuontityökalu.

Vanhan tietokannan tuonti vaatii uudelle palvelimelle tehtäviä muutoksia. PHP:n asetuksissa on oletuksena rajoitettu tuotavien tietokantojen koko kahteen megatavuun ja tuotavan tietokannan ollessa yli 68 megatavua täytyy rajaa korottaa. Korotus tapahtuu muokkaamalla PHP:n asetustiedostoa ”/etc/php5/apache2/php.ini” ja siellä muokataan ”upload_max_filesize”-kohdan määritystä kahdesta megatavusta esimerkiksi 80 megatavuun. Tämän jälkeen tuonti onnistuu.

Tuontivaiheessa ongelmaksi muodostuvat tuotavassa SQL-tiedostossa olevat kirjoitusvirheet. Kun tietokanta tuodaan, sen sisällä oleva SQL-koodi suoritetaan ja toimenpiteen aikana luodaan tietokannoille tarvittavat taulut ja käyttäjätunnukset. Koodissa on tarkkaa, että isot ja pienet kirjaimet ovat oikein. Tuotava tiedosto sisältää tässä tapauksessa virheitä kirjainten koossa ja nämä virheet täytyy käsin korjata ennen kuin koodi voidaan suorittaa onnistuneesti alusta loppuun. Korjaaminen suoritetaan yksinkertaisesti avaamalla SQL-tiedosto esimerkiksi Windowsin muistiolla ja muokkaamalla phpMyAdminin ilmoittamat virhepaikat.

Vanhan tietokannan tuominen uuteen palvelimeen aiheuttaa myös toisen odottamattoman ongelman. SQL-koodin luodessa käyttäjätunnuksia se samalla korvaa uudella tietokantapalvelimella jo valmiina olevia käyttäjätunnuksia ja salasanoja omillaan, jolloin tunnukset ja salasanat eivät enää välttämättä täsmää, ja esimerkiksi root-käyttäjätunnuksen kohdalla se aiheuttaa ongelmia. Ongelmat ilmenevät palvelimen käynnistyksen yhteydessä näkyvissä virheilmoituksissa, joissa ilmoitetaan käyttäjätunnusten tietokantaan kirjautumisen olevan estetty. Ongelma korjataan luomalla käyttäjätunnuksille uudet salasanat jolloin ne korvaavat myös tietokannassa olevat salasanat aiemmin käytössä olleilla. Salasanan vaihto tehdään palvelimella SQL-koodilla ”mysqladmin -u root password "**********"”, jossa tähdillä merkitty kenttä on uusi salasana.

Palvelimen käynnistyttyä onnistuneesti ilman virheilmoituksia voidaan se siirtää julkiseen verkkoon testikäyttöä varten. Tietohallinnon antama palvelimelle varattu IP-osoite otetaan käyttöön palvelimen kiinteäksi osoitteeksi, jolloin se ei enää käynnistyksen yhteydessä yritä saada uusia verkkoasetuksia. Verkkoasetukset vaihdetaan muokkaamalla tiedostoa ”/etc/network/interfaces” ja korvaamalla siellä olevat asetukset staattisilla osoitteilla. Tämän

jälkeen vSpheressä vaihdetaan palvelimen verkkoasetukset DMZ-asetuksia vastaaviksi ja palvelimen uudelleenkäynnistys siirtää sen julkiseen verkkoon ja valmiiksi käyttöön.

7 POHDINTA

Tämä opinnäytetyö on perusteiltaan selostus siitä, kuinka www-palvelin pystytetään. Se, että toisena osapuolena tiedonsiirrossa on ajoneuvo, ei muuta tilannetta mitenkään tekniikan toimiessa kuitenkin standardien mukaisesti. On siis käytännössä merkityksetöntä mitä laitteita asiakaspäässä on.

Tähän samaan ajatukseen pilvipalvelutkin perustuvat. Kun aiemmin ihmiskunta tallensi käytettäväksi kaiken tarvitsemansa datan itse paikallisesti omille laitteilleen, on nykyteknologia mahdollistanut tiedon tallentamisen fyysisesti kauas, mutta samalla siihen pääsee käsiksi millä tahansa laitteistolla. On kätevää tallentaa kotitietokoneelta musiikkia pilvipalvelimelle ja samalla kuunnella samaa musiikkia puhelimen välityksellä samalta palvelimelta. Sama musiikki soi, mutta alusta on eri. Tämän pilviteknologia mahdollistaa ja tämä on sen kantava idea.

Pilviteknologia ei siten todellisuudessa tuo mitään uutta. Tietoa on tallennettu ja käytetty eri tavoin jo vuosituhansia, mutta se mikä on muuttunut, on sen saatavuus. Nykyihmisen ajatusmaailmaan kuuluu, että kaikki tärkeä on heti oltava saatavilla ja aina täytyy olla tavoitettavissa. Mikäs silloin on sen parempaa kuin se, että ottaa mukaan mitä tahansa laitteita, niin aina ovat samat tiedot tallessa ja heti käden ulottuvilla.

Kuten Internetissä yleensäkin, ilmaista ateriaa ei ole olemassa. Harva osaa tehdä pilvipalvelua itse ja tällöin tukeudutaan jo olemassa oleviin palveluihin. Liitytään palveluun ja siirretään sinne omia, ehkä jopa tärkeitä yksityisiä tietoja ilman mitään epäilyä mistään. Onhan se todella epätodennäköistä, että tunnetuimmat pilvipalvelut uhkaisivat vakavasti tietoturvaa, mutta nykymaailmassa varmuus ei ole ehtymätön luonnonvara.

Kuten pohdinnan alussa mainittiin, on tämä opinnäytetyö www-palvelin, mutta silti puhutaan pilvipalveluista. Tämä on esimerkki siitä, että aiemmin tiettyihin lokeroihin kategorioituja palveluja voidaan nykymittapuulla arvioida uudelleen. Nyt rakennettu palvelin on etäkäyttöinen: Se suorittaa, käsittelee ja säilöö tietoa, jolloin asiakaskoneen tehtävänä on vain joko tarjota uutta käsittelemätöntä tietoa tai vastaanottaa käsiteltyä. Aivan siis kuten monessa pilvipalveluissa tapahtuu muutenkin.

Projektina palvelimen rakennus oli mielenkiintoinen. Se tarjosi monia odottamattomia ongelmia ja opetti käytännön palvelintekniikasta enemmän kuin teoriakurssit pystyisivät ikinä tarjoamaan. Suurimmat etenemistä haitanneet ongelmat kulminoituvatkin itse asiassa palvelimen ulkopuolelle.

Ensinnäkin Linuxin käytöstä komentorivipohjaisena ei ollut paljoa kokemusta. Tämä aiheutti sen, että alussa ei ollut aina tietoa siitä mitä pitäisi tehdä puhumattakaan siitä, että pitäisi myös tietää miten tehdään. Asialle ei oikeastaan voi muuta kuin ottaa selvää ja kysyä henkilöiltä jotka tietävät. Epäonnistuminen on avain onnistumiseen, ennemmin tai myöhemmin.

Vaikka asiaa kuinka kääntelee, niin totuus on että Linux on niin sanottu nörttikäyttöjärjestelmä. Tämän huomaa erityisesti siinä, että vaikka ohjeistusta on runsaasti tarjolla, on se monesti kohdistettu sellaisille henkilöille joiden lähtötaso on valmiiksi korkea. Ohjeissa oletetaan lukijansa tietämyksen olevan usein korkea ja tämä taas aiheuttaa monelle pään iskeytymistä seinään.

Ehkä tärkein asia rakennettaessa jotain itselle uutta ja haastavaa on pitää henkinen puoli kunnossa. On helppoa heittää pyyhe kehään ja jättää työ kesken, mutta silloin ei voi myöskään voittaa yhtään mitään. Itsensä ylittäminen mahdollistaa sen, että kun siitä eteenpäin rima menee aina vain korkeammalle ja samalla myös onnistumiset kasvavat.

Lopputulos itsessään on ensikertalaiselle varsin kelpo. Ei se varmastikaan ole täydellinen, mutta aina sitä voi korjata jälkikäteen. Kokonaisuudessaan voikin todeta, että tärkeintä ei ole määränpää, vaan se matka.

LÄHTEET

Abrams, L. 2004. TCP and UDP Ports Explained. Saatavilla: http://www.bleepingcomputer.com/tutorials/tcp-and-udp-ports-explained/ (Luettu 18.5.2012).

Apache http server project. 2012. Saatavilla: http://httpd.apache.org/ (Luettu 24.4.2012).

Byfield, B. Linux Planet. 2009. Why Debian is the Leader of the Linux Pack. Saatavilla: http://www.linuxplanet.com/linuxplanet/reviews/6677/1 (Luettu 9.5.2012).

DifferenceBetween.net. 2012. Difference Between Ubuntu Desktop and Server. Saatavilla: http://www.differencebetween.net/technology/difference-between-ubuntu-desktop-and-server/ (Luettu 9.5.2012).

Emspak, J. Discovery News. 2011. Fords Evos Takes Cars to the Cloud. Saatavilla: http://news.discovery.com/autos/ford-evos-cloud-computing-car-110913.html (Luettu 9.5.2012).

Garner, A. LinuxSA. 2012. Linux Tips Disk Partitioning. Saatavilla: http://www.linuxsa.org.au/tips/disk-partitioning.html (Luettu 9.5.2012).

Heinonen, P. Jyväskylän yliopisto. 2002. Tiedonsalaaminen. Saatavilla: http://appro.mit.jyu.fi/doc/tiedonsalaus/ (Luettu 2.5.2012).

Hess, K. 2010. Ten Reasons to Dumb Windows and Use Linux. Saatavilla: http://www.pcworld.com/businesscenter/article/201731/ten_reasons_to_dump_windows_and_use_linux.html (Luettu 16.4.2012).

Linux.org. 2012. What is Linux?. Saatavilla: http://www.linux.org/article/view/what-is-linux (Luettu 10.5.2012).

Markoff, J. The New York Times. 2010. Smarter Than You Think – Google Cars Drive Themselves, in Traffic. Saatavilla: http://www.nytimes.com/2010/10/10/science/10google.html?pagewanted=1&_r=1 (Luettu 9.5.2012).

Mell, P. & Grance, T. National Institute of Standards and Technology. 2011. The NIST Definition of Cloud Computing. Saatavilla: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (Luettu 9.5.2012).

Oracle. 2012. Why MySQL?. Saatavilla: http://www.mysql.com/why-mysql/ (Luettu 9.5.2012).

Petri, D. 2009. Understanding Windows Server 2008 Server Core. Saatavilla: http://www.petri.co.il/understanding-windows-server-2008-core.htm (Luettu 9.5.2012).

Pettey, C. Gartner. 2011. Gartner Says Worldwide Operating System Software Market Grew to $30.4 Billion in 2010. Saatavilla: http://www.gartner.com/it/page.jsp?id=1654914 (Luettu 16.4.2012).

Repo, H. Tekniikka & Talous. Sonera tyynnyttelee verkon tukkeutumispuheita – Ruotsissa 3G-ongelmat nousseet hallitukseen asti?. Saatavilla: http://www.tekniikkatalous.fi/ict/sonera+tyynnyttelee+verkon+tukkeutumispuheita++ruotsissa+3gongelmat+nousseet+hallitukseen+asti/a718598 (Luettu 18.5.2012).

Smith, R.W. 2009. LPIC-1 Linux Professional Institute Certification Study Guide. Indianapolis: Wiley Publishing, Inc.

Squatriglia, C. Wired. 2012. Cars connect With Apps, the Cloud at CES. Saatavilla: http://www.wired.com/gadgetlab/2012/01/cars-connect-with-apps-the-cloud-at-ces/all/1 (Luettu 9.5.2012).

Symantec. 2012. Secure Sockets Layer (SSL): How It Works. Saatavilla: https://www.symantec.com/theme.jsp?themeid=how-ssl-works (Luettu 2.5.2012).

The PHP Group. 2012. FAG. Saatavilla: http://fi2.php.net/manual/en/faq.general.php (Luettu 9.5.2012).

Torikka, M. 2009. Android tekee pingviinistäkin seksikkään. Tekniikka&Talous 20.2.2009, 14-15.

Tuxradar.com. 2009. How to choose the best Linux distro for you. Saatavilla: http://www.tuxradar.com/content/how-choose-best-linux-distro (Luettu 18.5.2012).

Unuth, N. About.com. 2012. 3G – What is 3G?. Saatavilla: http://voip.about.com/od/mobilevoip/p/3G.htm (Luettu 18.5.2012).

Venezia, Paul. PC World. 2010. How to set up a virtual server. Saatavilla: http://howto.techworld.com/virtualisation/3232754/how-to-set-up-a-virtual-server/?pn=1 (Luettu 18.5.2012).

Viestintävirasto. 2011. Ohjeita matkapuhelimen kuuluvuuden ja mobiililaajakaistan toimivuuden parantamiseksi. Saatavilla: http://www.viestintavirasto.fi/index/puhelin/ohjeitakuuluvuudenparantamiseen.html#s (Luettu 18.5.2012).

Vlaurie.com. 2012. Ports and the Security of Your Internet Connection. Saatavilla: http://vlaurie.com/computers2/Articles/ports.htm (Luettu 18.5.2012).

Digi International. 2009. Efficient Data Transfer over Cellular Networks. White Paper A2/309, 5. Saatavilla: http://www.digi.com/pdf/wp_efficientdatatransfer.pdf

Wikipedia. 2012. HTTPS. Saatavilla: http://fi.wikipedia.org/wiki/HTTPS (Luettu 10.5.2012).